MODUL 1

PERANCANGAN STAR SCHEMA DAN SNOWFLAKE

©Muhammadiyah university Press

1.1. Tujuan

1. Mahasiswa mampu menjelaskan prosedur perancangan *Star Schema* atau *Snowflake*.
2. Mahasiswa mampu merancang *Star Schema* atau *Snowflake* menggunakan program aplikasi tertentu.

1.2. Landasan Teori

Apa hubungan antara *star schema* dan *snowflake* dengan basis data? Kedua istilah skema ini mewakili struktur basis data yang umum digunakan pada basis data OLAP (*On Line Analytical Processing*) untuk kebutuhan *data warehouse*. Dalam banyak pelajaran tentang basis data, kedua skema ini jarang disampaikan akibat penerapannya yang tidak sesuai untuk model basis data OLTP (*On Line Transactional Processing*). Mekanisme normalisasi juga tidak banyak berlaku untuk kedua jenis skema basis data ini. Fokus utama materi dasar-dasar basis data adalah proses manipulasi data, dalam hal ini bagaimana merancang sistem basis data yang dapat melayani sekian transaksi DML mulai *Insert*, *Update*, dan *Delete*? Bagaimana melakukan normalisasi pada struktur basis data untuk mendapatkan struktur yang ideal? Bagaimana mengatur transaksi antar klien agar tidak muncul *deadlock*? Dan banyak pertanyaan yang muncul terkait dengan sistem basis data.

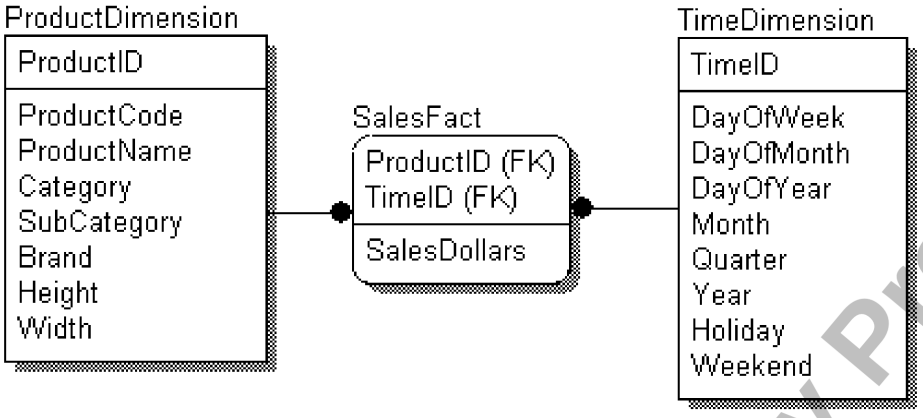
Struktur data pada OLAP jauh lebih sederhana, mengingat data-data yang akan tersimpan di dalamnya tidak banyak mengalami perubahan dimana lebih banyak transaksi *selection* (*read only* – hanya baca) daripada DML. Jika pada OLTP, konsep ACID (*atomicity, consistency, isolation, durability*) menjadi properti utama yang harus melekat pada setiap transaksi data dari dan ke aplikasi klien maka dalam OLAP yang lebih diutamakan adalah kecepatan perolehan datanya (*data retrieval*). Tidak hanya struktur basis datanya yang berbeda, namun konfigurasi server basis datanya pun akan berbeda antara OLAP dan OLTP.

1.2.1. *Star Schema*

Dalam *data warehouse*, data-datanya akan disimpan dalam tabel fakta dan tabel dimensi. Tabel fakta akan menyimpan data-data utama sementara tabel dimensi mendeskripsikan setiap nilai dari suatu dimensi dan dapat direlasikan ke tabel fakta jika diperlukan. Data fakta merupakan data yang terukur besarannya, sebagai contoh adalah jumlah siswa, banyaknya rupiah yang diperoleh, rata-rata IPK, dan sejenisnya. Untuk lebih menjelaskan data fakta, maka kondisi saat data tersebut diukur turut disampaikan. Data kondisi inilah yang dipetakan dalam bentuk data dimensi. Kondisi yang dipetakan dalam dimensi umumnya berupa kondisi waktu, kondisi produk atau item, dan kondisi geografisnya. Mendesain struktur *star schema*, dimulai dengan menentukan data apa yang ingin dilihat oleh pengguna (besarannya) dan bagaimana pengguna melihat data tersebut (kondisi atau dimensinya).

Tabel dimensi memiliki *primary key* sederhana yang mengandung hanya satu atau dua kolom saja. Namun, tabel fakta akan memiliki sekumpulan *foreign key* yang disusun dari *primary key* komposit dan merupakan gabungan kolom- kolom tabel dimensi yang berelasi. Untuk lebih jelasnya, berikut contoh struktur *star schema*.

©Muhammadiyah University Press



Gambar 1.1 Contoh *Star Schema*

Untuk struktur *star schema* seperti Gambar 1.1, data dalam tabel fakta yang diukur adalah hasil penjualan (dalam mata uang dollar) berdasarkan dimensi atau kondisi produk yang dijual (*product*) serta waktu penjualan (*time*). Misalkan dimensi produk, yang menyimpan informasi-informasi seputar produk. Produk ini dapat dikelompokkan ke dalam kategori, dan di dalam kategori inipun bisa ditemukan sub-kategori. Misalkan dalam sebuah basis data terdapat kode produk **X1001** yang merujuk pada kripik tempe, maka akan masuk ke dalam kategori Nabati, dan sub-kategori Tempe. Untuk lebih mengelompokkan produk tersebut, dapat pula dibuatkan sub-kategori berikutnya. Namun kunci dari informasi produk tersebut tersimpan dalam kolom di tabel dimensi, dan tidak dibutuhkan tabel lain untuk menjelaskan detil produk. Semakin beragam jenis kondisi data yang ingin diamati, maka akan semakin besar ukuran tabel fakta yang dimuat.

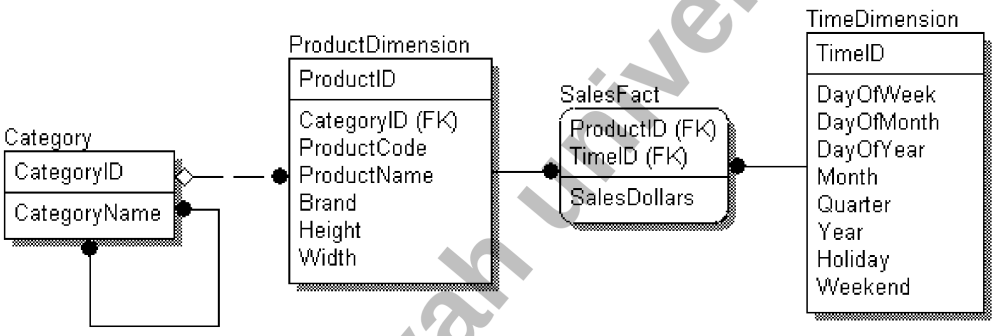
Dalam *star schema*, *query* yang terbentuk antara tabel fakta dan sejumlah tabel dimensi dinamakan ***star query***. Setiap tabel dimensi direlasikan dengan tabel fakta berdasarkan kolom *primary key* dan *foreign key*, namun diantara masing- masing tabel dimensi tidak ada yang saling berelasi (tidak ada hubungan data). Query yang terbentuk menyebabkan proses eksekusi yang lebih optimal, karena rencana eksekusi query dalam DBMS akan lebih cepat dengan setiap tabel hanya berelasi dengan satu tabel yang lain.

©Muhammadiyah University Press

Ada kalanya tabel dimensi mengandung data yang duplikat pada satu atau lebih kolom. Jika mengikuti azas normalisasi, maka struktur basis data yang terbentuk bukan lagi *star schema* namun akan menjadi *snowflake schema*.

1.2.2. *Snowflake Schema*

Struktur basis data ini lebih kompleks dari pada *star schema*, dengan menormalisasi tabel-tabel dimensi yang berukuran besar dengan satu atau lebih kolom yang memiliki duplikasi data. Misalkan jika tabel dimensi *Product* dinormalisasi maka akan menghasilkan struktur seperti berikut:



Gambar 1.2 Contoh Bentuk *Snowflake*

Tabel dimensi dinormalisasi untuk mengurangi redudansi data (duplikasi), sehingga struktur tabelnya akan lebih ramping. Dengan pengelompokkan ini, data akan lebih mudah dibaca dan membantu pengembang aplikasi untuk menata desain antarmuka sistem dan *filtering* data. Struktur ini akan menghemat kapasitas *storage*, namun waktu eksekusi data akan lebih lama mengingat jumlah tabel dimensi yang direlasikan lebih banyak dan membutuhkan tambahan relasi *foreign key*. *Query* yang terbentuk lebih kompleks, yang mengakibatkan kinerja *query* menurun. Pada penerapan yang lebih umum, tabel dimensi tidak diturunkan dengan lebih banyak tabel dimensi lain dan pengelompokan data diatur secara *hard-coded* di kode program aplikasinya.

©Muhammadiyah University Press

Fokus penggunaan *data warehouse* adalah kecepatan akses dan eksekusi data, bukanlah ukuran data yang lebih kecil atau struktur basis data yang lebih ramping. Sehingga bijaksana dalam menetapkan struktur data *star* maupun *snowflake schema* akan menentukan kinerja layanan *data warehouse* yang dimiliki.

Tahap pertama dari perancangan *data warehouse* adalah mendefinisikan informasi-informasi apa saja yang dibutuhkan oleh manajemen. Agar kebutuhan ini dapat didefinisikan dengan tepat, maka pemahaman akan peran dan tugas manajemen yang membutuhkan informasi tersebut mutlak harus dilakukan lebih dulu. Jika sudah dipahami, selanjutnya kita hanya tinggal "menjawab" pertanyaan-pertanyaan berikut:

1. Siapa yang membutuhkan informasi dari *data warehouse*?
2. Informasi apa saja yang dibutuhkan tersebut?
3. Seperti apa *layout* dan isi informasi-informasi itu?
4. Kapan informasi tersebut digunakan?
5. Untuk keperluan apa?
6. Basis data apa yang menjadi sumber untuk informasi tersebut?

Sebagai contoh, misalkan akan dibuat sebuah data warehouse penjualan (atau data mart penjualan tepatnya) untuk sebuah perusahaan dagang.

1. Siapa yang membutuhkan informasi dari data warehouse?
   Manager Pemasaran
2. Informasi apa saja yang dibutuhkan Manager Pemasaran?
   Barang apa yang paling banyak terjual di lokasi tertentu sepanjang tahun?
   Barang apa yang paling banyak memberikan pendapatan sepanjang tahun?
3. Seperti apa *layout* dan isi informasi-informasi itu?
   Barang yang paling banyak terjual di lokasi tertentu sepanjang tahun:

©Muhammadiyah University Press

tahun	kecamatan	kategori	sum(total_penjualan)
2012	BANJARSARI	KONSUMSI	209
2012	JEBRES	ATK	95
2012	LAWEYAN	ATK	109
2012	SERENGAN	ATK	89
2012	JEBRES	KONSUMSI	106
2012	PASAR KLIWON	KONSUMSI	96
2012	BANJARSARI	ATK	200
2012	LAWEYAN	KONSUMSI	193
2012	PASAR KLIWON	ATK	91
2012	SERENGAN	KONSUMSI	139

Barang yang paling banyak memberikan pendapatan sepanjang tahun:

tahun	kategori	sub_kategori	sum(total_penerimaan)
2012	ATK	KERTAS	3560000
2012	ATK	PULPEN	472000
2012	ATK	SPIDOL	1269000
2012	KONSUMSI	SEMBAKO	524000
2012	KONSUMSI	SNACK	1669500

4. Untuk keperluan apa informasi tersebut?
   Dasar untuk menentukan strategi penjualan barang
5. Kapan informasi tersebut digunakan?
   Awal periode penjualan
6. Basis data apa yang menjadi sumber untuk informasi tersebut?
   Basis data penjualan dengan skema sebagai berikut:
   a) Kategori (#kelompok, sub_kategori, kategori)
   b) Barang (#kode_barang, nama_barang, #kelompok, satuan, harga)
   c) Lokasi (#kode_pos, kelurahan, kecamatan)
   d) Pelanggan (#kode_pelanggan, nama_pelanggan, alamat, kota, #kode_pos, telepon)

©Muhammadiyah university Press

e) Penjualan (#no_faktur, #kode_barang, jumlah)
f) Pembayaran (#no_faktur, tanggal, total, diskon, #kode_pelanggan)

Tahap berikutnya yang harus dilakukan adalah menentukan ***measure*** dan ***dimension*** untuk semua informasi yang dibutuhkan manajemen. ***Measure*** adalah data numerik yang akan dicari jejak nilainya, sedangkan ***dimension*** adalah parameter atau sudut pandang terhadap *measure* sehingga dapat mendefinisikan suatu transaksi.

Sebagai contoh, untuk informasi "barang yang paling banyak terjual di lokasi tertentu sepanjang tahun",

1. *Measure*: total penjualan
2. *Dimension*: barang, tahun (waktu/periode), lokasi

Sedangkan untuk informasi "barang yang paling banyak memberikan pendapatan sepanjang tahun",

1. *Measure*: total pendapatan
2. *Dimension*: barang, tahun (waktu/periode)

Dimension mempunyai hirarki. Penentuan hirarki untuk *dimension* ini sepenuhnya tergantung kepada proses *drill down* dan *roll up* yang ingin dilakukaan saat melakukan OLAP (*On Line Analythical Processing*) nanti.

Untuk contoh di atas, hirarki masing-masing *dimension* adalah:

1. Barang: nama barang, sub-kategori, kategori
2. Periode: minggu, bulan, tahun
3. Lokasi: kelurahan, kecamatan, kota

Sedangkan *layout* dan isi informasinya dapat ditunjukkan oleh tabel berikut ini:

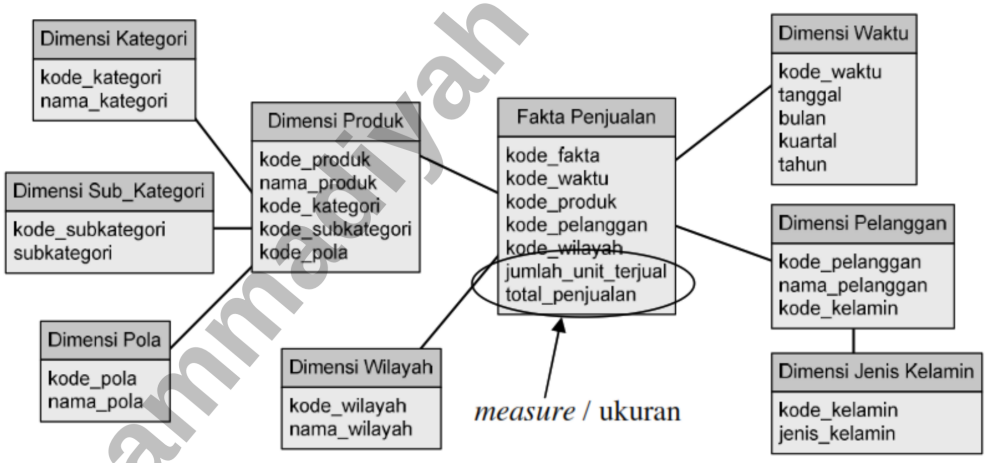
©Muhammadiyah university Press

BARANG				PERIODE			LOKASI			TOTAL PENJUALAN	TOTAL PENERIMAAN
KODE	NAMA	SUB KATEGORI	KATEGORI	MINGGU	BULAN	TAHUN	KELURAHAN	KECAMATAN	KOTA		
SP-001	BOARDMARKER SNOWMAN	SPIDOL	ATK	33	8	2012	JEBRES	JEBRES	SURAKARTA	12	Rp 78,000
				38	9	2012	GAJAHAN	PASAR KLIWON	SURAKARTA	7	Rp 45,500
				43	10	2012	KAUMAN	PASAR KLIWON	SURAKARTA	7	Rp 45,500
				7	2	2012	GILINGAN	BANJARSARI	SURAKARTA	12	Rp 78,000
										38	Rp 247,000
PL-002	PULPEN PILOT	PULPEN	ATK	22	6	2012	DANUKUSUMAN	SERENGAN	SURAKARTA	3	Rp 9,000
				26	6	2012	KARANGASEM	LAWEYAN	SURAKARTA	6	Rp 18,000
				41	10	2012	KERTEN	LAWEYAN	SURAKARTA	8	Rp 24,000
										17	Rp 51,000
KH-005	A4 SINAR DUNIA	KERTAS	ATK	10	3	2012	MOJOSONGO	JEBRES	SURAKARTA	10	Rp 350,000
				27	7	2012	TIPES	SERENGAN	SURAKARTA	4	Rp 140,000
				46	11	2012	MANAHAN	BANJARSARI	SURAKARTA	9	Rp 315,000
										23	Rp 805,000
JUMLAH										78	Rp1,103,000

©Muhammadiyah University Press

Perancangan model konseptual data warehouse adalah tahap berikutnya yang harus dilaksanakan setelah tahap penentuan *measure* dan *dimension*. Pada tahap ini dibuat suatu model yang dapat menggambarkan data atau tabel apa saja yang akan disimpan dalam *data warehouse*, berikut keterhubungan diantaranya.

Data atau tabel dalam *data warehouse* tersebut dapat dimodelkan dengan menggunakan alat bantu pemodelan seperti E-R diagram, *star schema, snowflake schema,* atau FCO-IM (*Fully Communication Oriented Information Modelling*). Tetapi pada umumnya alat bantu yang digunakan adalah *star schema* atau *snowflake schema*. *Star schema* digunakan untuk menggambarkan *fact table*, yaitu tabel yang merepresentasikan *measure*, sebagai "pusat data". Tabel ini nantinya akan terkoneksi dengan tabel-tabel yang mendeskripsikan dimensi untuk *measure* tersebut (*dimension table*). Sebagai contoh, *snowflake schema* untuk *data warehouse* penjualan di sebuah perusahaan batik adalah:



Gambar 1.3 *Snowflake Schema* Penjualan

1.3. Alat dan Bahan

1. Komputer dengan sistem operasi Windows.
2. Program aplikasi DBDesigner.
3. Modul Praktikum Data Warehousing dan Data Mining.

©Muhammadiyah university Press

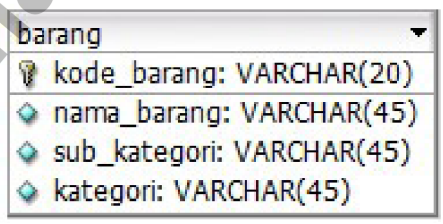
1.4. Langkah-langkah Praktikum

Menggambar *Star Schema* dengan menggunakan DB Designer:

1. Jalankan program aplikasi DB Designer untuk membuat desain *star schema*.
2. Klik button *new table* kemudian klik pada area kerja sehingga akan menghasilkan tabel baru.
3. Double klik pada tabel baru untuk membuka tabel editor, ganti nama pada *table name* dengan nama **barang**, kemudian isikan atribut tabel dengan data sebagai berikut :

Column Name	Data Type
kode_barang	Varchar(20)
nama_barang	Varchar(45)
sub_kategori	Varchar(45)
kategori	Varchar(45)

4. Klik pada *column name* kode_barang untuk mengatur kode_barang sebagai *primary key* sehingga berubah menjadi .
5. Klik untuk menutup *table editor* sehingga tabel barang menjadi :



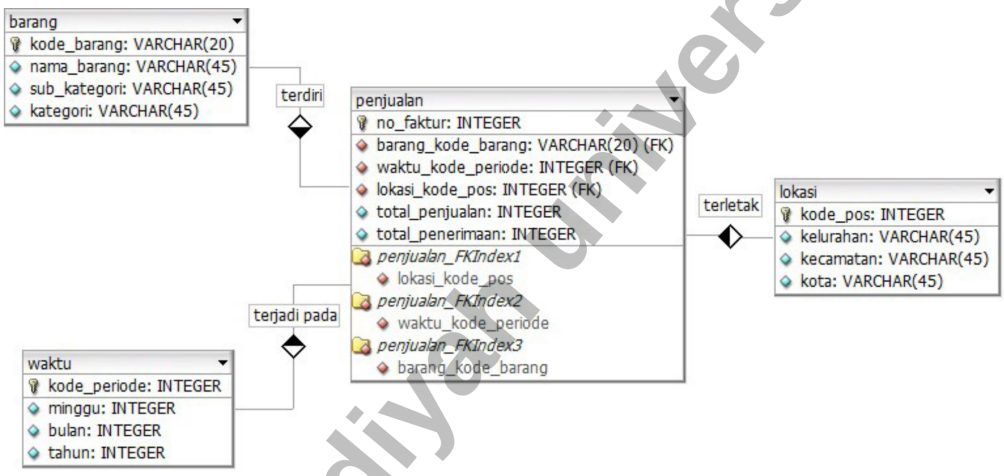
6. Ulangi kembali langkah 2 sampai 5 untuk membuat tabel **waktu**, **lokasi** dan **penjualan**.
7. Setelah semua tabel dibuat, hubungkan setiap tabel dengan tabel lain dengan button sebagai berikut :

©Muhammadiyah University Press

Button	Fungsi Relationship
	1:n (*one to many*)
	1:1 (*one to one*)
	n:m (*many to many*)

Keterangan: klik salah satu button yang sesuai dengan kebutuhan kemudian klik tabel yang akan dihubungkan.

8. Ubah nama *relationship* dengan membuka *relationship editor*, sehingga setelah selesai hasil akhir manjadi seperti berikut:



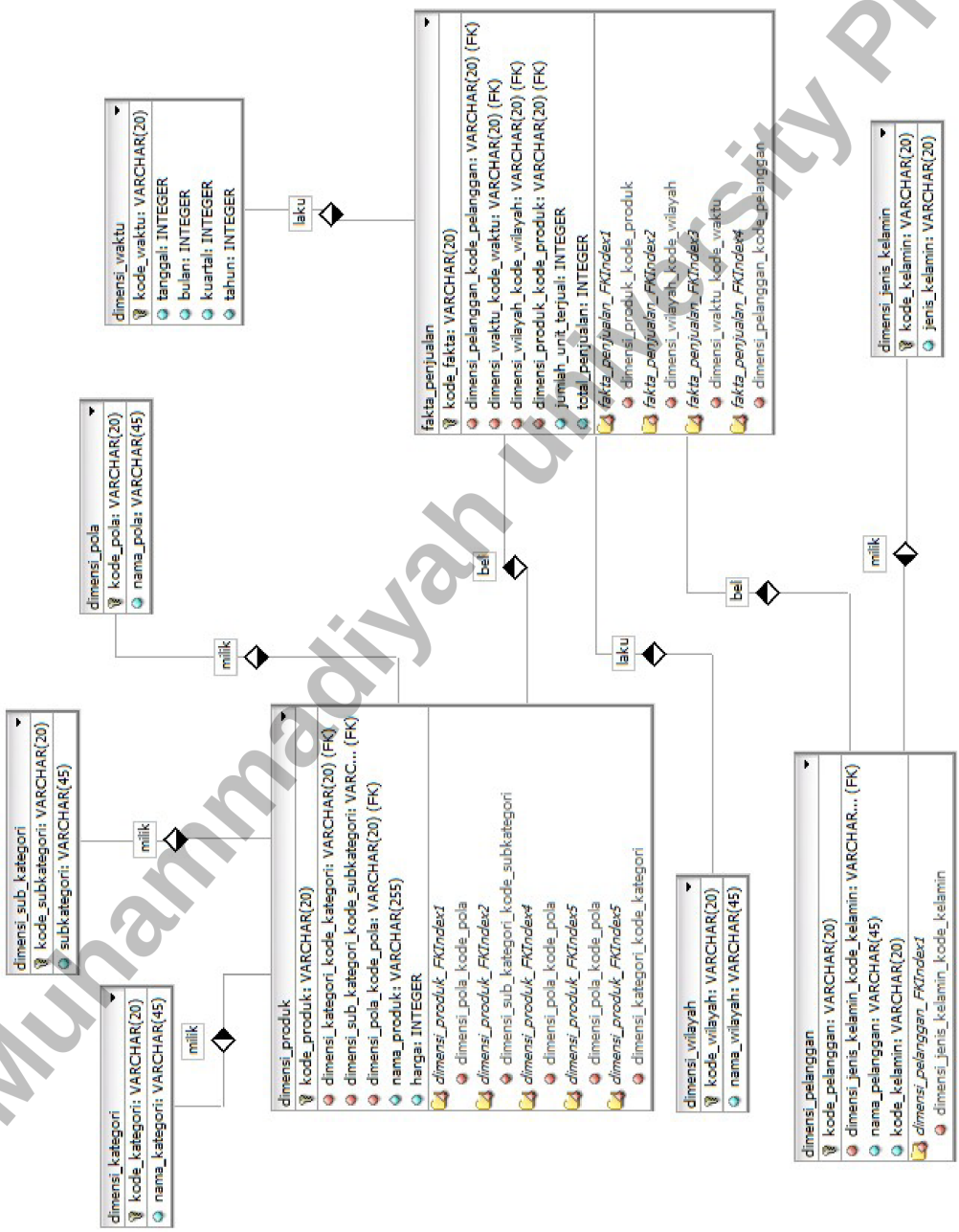
9. Simpan dengan nama file "**star schema penjualan.xml**" dalam folder "Praktikum Data Warehousing dan Data Mining"

Snowflake schema merupakan perbaikan dari *star schema*, sehingga cara penggambarannya pun mirip. Bedanya, satu atau beberapa hirarki yang ada pada *dimension table* dinormalisasi (dekomposisi) menjadi beberapa tabel yang lebih kecil.

©Muhammadiyah University Press

1.5. Tugas

1. Rancanglah diagram *Snowflake schema* berdasarkan gambar di bawah dengan menggunakan DBDesigner seperti gambar berikut! Simpan dengan nama file ”**snowflake penjualan.xml**” ke dalam folder ”Praktikum Data Warehouse”



©Muhammadiyah University Press